CLAIMS

1. A method of monitoring response time of a method or a function associated with a Java software component, comprising the steps of:

inserting an instrumentation code in a bytecode representation of said method or function to effect generation of a start time marker upon start of execution of said method or function and a stop time marker upon completion of execution of said method or function, and

utilizing said start and stop time markers to determine a response time of said method or function.

2. The method of claim 1, wherein said inserted instrumentation code generate calls to an Application Response Measurement (ARM) agent to cause the agent to provide said start and said stop time markers.

3. The method of claim 2, further comprising:

registering said method or function with said ARM agent prior to generation of said start and stop time markers.

4. The method of claim 2, wherein said instrumentation code causes generation of said start and stop time markers without modifying instructions associated with execution of said method or function.

5. The method of claim 2, wherein said ARM agent generates a record corresponding to said method or function for storing the response time associated with said method or function.

6. The method of claim 5, wherein said record includes a field for identifying a parent, if any, of said method or function in a hierarchical parent-child transaction chain.

7. The method of claim 6, wherein said record includes another field for identifying a top level transaction in said parent-child transaction chain.

8. The method of claim 1, wherein said software component can be any of a JSP, a servlet a EJB, a JDBC driver, a JNDI or an RMI component.

9. The method of claim 8, wherein said method of function of the software component comprises any of a _jspService method of a JSP, a doFilter, a doGet, a doPost or a service method of a servlet, a getConnection, executeQuery, or selected methods of JDBC driver, or remote, local or home interface methods of an EJB..

10. The method of claim 1, wherein the step of inserting the instrumentation code comprises incorporating instrumentation hooks into said bytecode representation prior to or during loading and initialization of a class containing said method or function by a Java Virtual Machine (JVM).

11. The method of claim 1, further comprising:

storing said response time in a database.

12. The method of claim 1, further comprising:

displaying said response time to a user.

13. A system for monitoring response time of a method or function associated with a Java software component, comprising:

an instrumentation engine for inserting instrumentation code in a byte code representation of said method or function, said instrumentation code effecting generation of a start time marker and a stop time marker upon resumption and completion, respectively, of said method or function,

an interface module being invoked by said instrumentation code upon start and completion of said method or function, and

an application response measurement (ARM) agent in communication with said interface module,

wherein said interface module, upon invocation by said instrumentation code, calls said ARM agent to cause generation of said start and stop time markers by said ARM agent.

14. The system of claim 13, wherein said instrumentation engine inserts said instrumentation code prior to loading of a class containing said method or function by a Java Virtual Machine (JVM).

15. The system of claim 13, wherein said instrumentation engine inserts said instrumentation code in said bytecode representation without modifying instructions associated with execution of said method or function.

16. The system of claim 13, further comprising:

an analysis and presentation module in communication with said ARM agent for presenting said response time to a user and/or storing said response time in a database.

17. A method of measuring response time of a transaction initiated by a web server in response to an HTTP request, comprising

utilizing a call back provided by the web server to save a start time marker upon start of said transaction,

utilizing another call back from the web server to save a stop time marker upon completion of said transaction, and

utilizing said start and stop time markers to measures a response time associated with said transaction.

18. The method of claim 17, further comprising

deploying a monitoring agent on said web server for receiving said call backs and generating said start and stop time markers.

19. The method of claim 18, further comprising

registering said monitoring agent with said web server to receive said call backs.

20. The method of claim 19, further comprising

transmitting said response time to a presentation module for presentation to a user.

21. A method of measuring a response time associated with a request transmitted from a web browser to a web server, comprising

deploying a script entity on said web browser, said script entity registering with a web server to receive one or more call backs therefrom at selected points during a transaction initiated by said web server in response to the request from the browser,

utilizing said script entity to start a clock upon transmission of said request from said browser to said web server,

utilizing said script entity to stop the clock upon receipt of a call back from said server indicating completion of said transaction.